IN THE CLAIMS:

Please amend Claims 1 and 2 as follows:

1. (amended) In a system for digital information processing [process], said system having a memory, a method for generating data representative of [finding] a quotient $Q = a_0a_1a_2....a_b$ from data representative of a divisor $Y = y_1y_2....y_n$ and data representative of a dividend $X = x_1x_2....x_a$, comprising the [following] steps of:

(a) aligning the first non-zero bit of X with the first non-zero digit of Y;

(b) defining a signed-digit partial remainder series R_i where $R_0 = Y$, a first sign series of the partial remainder S_i where $S_0 = 0$, a second sign series of the partial remainder S_{ri}, a quotient bit series a_i, and a counter i beginning from zero;

(c) subtracting X from R_i which yields next signed-digit partial remainder R_{i+1};

(d) setting the sign of R_{i+1} to S_{ri+1};

(e) setting the result of exclusive-OR of S_i and S_{ri+1} to the true sign of the next remainder S_{i+1};

(f) setting a_i to 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;

(g) setting a_i to 0 if $S_{i+1} = 1$;

(h) inverting the signs of all digits of R_{i+1} if $S_{i+1} = 1$;

22

(i) shift R_{i+1} left by one bit;

(j) adding 1 to i; [and]

(k) repeating [said] steps (c) to (j) until i reaches a predetermined value or $R_{i+1} = 0$[.]; and

(l) storing in said memory as said data representative of a quotient, a quotient resulting from step (k).

2. (amended) In a system for digital information processing [process], said system having a memory for storing data , a method for generating data representative of [finding] a signed magnitude quotient $Q_2 = a_s a_0.a_1 a_2....a_b$ from data representative of a signed divisor $Y_s = y_s.y_1 y_2....y_n$, , and data representative of a signed dividend $X_s = x_s.x_1 x_2....x_s$, comprising the [following] steps of:

(a) obtaining a_s from the result of exclusive-OR of y_s and x_s;

(b) defining a divisor $Y = y_1 y_2....y_n$, a dividend $X = x_1 x_2....x_s$, a signed-digit partial remainder series R_i where $R_0 = Y$, a first sign series of the partial remainder S_i where $S_0 = 0$, a second sign series of the partial remainder S_{ri}, a quotient bit series a_i, and a counter i beginning from zero;

(c) aligning the first non-zero bit of X with the first non-zero digit of Y;

(d) subtracting X from R_i which yields next signed-digit partial remainder R_{i+1};

(e) setting the sign of R_{i+1} to S_{ri+1};

(f) setting the result of exclusive-OR of S_i and S_{ri+1} to the true sign of the next remainder S_{i+1};

(g) setting a_i to 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;

(h) setting a_i to 0 if $S_{i+1} = 1$;

(i) inverting the signs of all digits of R_{i+1} if $S_{i+1} = 1$;

(j) shift R_{i+1} left by one bit;

(k) adding 1 to i; and

(l) repeating [said] steps (d) to (k) until i reaches a predetermined value or $R_{i+1} = 0$ [.]; and

(l) storing in said memory as said data representative of a signed magnitude quotient, a quotient resulting from step (k).

Please add new Claims 3 and 4 as follows:

3. (new) A system for digital information processing, said system having a memory for storing data, including data representative of a quotient $Q = a_0a_1a_2....a_b$ from data representative of a divisor $Y = y_1y_2....y_n$ and data representative of a dividend $X = x_1x_2....x_a$, said data representative of a quotient generated by a method comprising the steps of:

(a) aligning the first non-zero bit of X with the first non-zero digit of Y;

(b) defining a signed-digit partial remainder series R_i where $R_0 = Y$, a first sign series of the partial remainder S_i where $S_0 = 0$, a second sign series of the partial remainder S_{ri}, a quotient bit series a_i, and a counter i beginning from zero;

24

(c) subtracting X from R_i which yields next signed-digit partial remainder R_{i+1};

(d) setting the sign of R_{i+1} to S_{ri+1};

(e) setting the result of exclusive-OR of S_i and S_{ri+1} to the true sign of the next remainder S_{i+1};

(f) setting a_i to 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;

(g) setting a_i to 0 if $S_{i+1} = 1$;

(h) inverting the signs of all digits of R_{i+1} if $S_{i+1} = 1$;

(i) shift R_{i+1} left by one bit;

(j) adding 1 to i;

(k) repeating steps (c) to (j) until i reaches a predetermined value or $R_{i+1} = 0$; and

(l) storing in said memory as said data representative of a quotient, a quotient resulting from step (k).

4. (new) A system for digital information processing, said system having a memory for storing data, including data representative of a signed magnitude quotient $Q_2 = a_s a_0.a_1 a_2....a_b$ from data representative of a signed divisor $Y_s = y_s.y_1 y_2....y_n$, , and data representative of a signed dividend $X_s = x_s.x_1 x_2....x_s$, said data representative of a signed magnitude quotient generated by a method comprising the steps of:

(a) obtaining a_s from the result of exclusive-OR of y_s and x_s;

25

(b) defining a divisor $Y = y_1y_2....y_n$, a dividend $X = x_1x_2....x_s$, a signed-digit partial remainder series R_i where $R_0 = Y$, a first sign series of the partial remainder S_i where $S_0 = 0$, a second sign series of the partial remainder S_{ri}, a quotient bit series a_i, and a counter i beginning from zero;

(c) aligning the first non-zero bit of X with the first non-zero digit of Y;

(d) subtracting X from R_i which yields next signed-digit partial remainder R_{i+1};

(e) setting the sign of R_{i+1} to S_{ri+1};

(f) setting the result of exclusive-OR of S_i and S_{ri+1} to the true sign of the next remainder S_{i+1};

(g) setting a_i to 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;

(h) setting a_i to 0 if $S_{i+1} = 1$;

(i) inverting the signs of all digits of R_{i+1} if $S_{i+1} = 1$;

(j) shift R_{i+1} left by one bit;

(k) adding 1 to i;

(l) repeating steps (d) to (k) until i reaches a predetermined value or $R_{i+1} = 0$; and

(m) storing in said memory as said data representative of a signed magnitude quotient, a quotient resulting from step (l).